Top Informix-PostgreSQL Conversion "Gotchas" 9/20/2004

(1) In Informix, the normal file extension for files with C code with embedded SQL is ".ec". In PostgreSQL, it is ".pgc".

(2) Within an SQL statement, Informix denotes a string using double quotes ("). PostgreSQL denotes these strings using single quotes (').

Informix:

```
SELECT * FROM Location WHERE lid = "BLUO2";
```

PostgreSQL:

```
SELECT * FROM Location WHERE lid= 'BLUO2';
```

This is true for both ad hoc queries and .pgc files.

Note that character strings in C are still denoted using double quotes.

(3) The Informix utilities dbaccess and sqlcmd for ad hoc queries are replaced by a PostgreSQL utility called psql.

(4) PostgreSQL does not implement an ISAM error code.

(5) PostgreSQL error return codes -239 and -284 are the same as Informix. All other error codes are different. Code which tests Informix error return codes other than these two must be changed.

(6) ecpg, the PostgreSQL embedded C pre-compiler, is an executable which does not have a compile option. You must compile the code in a separate step. Informix's esql is a script which includes both a pre-compiler and compiler.

(7) "value" is a reserved word in PostgreSQL. Therefore statements such as

```
SELECT value FROM Precip …
```

result in an error and must be rewritten as

```
SELECT "value" FROM Precip …
```

(8) In ecpg, resources related to a cursor are freed via the CLOSE statement. Unlike esql, ecpg does NOT implement a FREE statement. Therefore statements such as

```
EXEC SQL FREE …
```

should be removed.

(9) When records are deleted from a table using a DELETE, the deleted records are not removed from the db and the free space is NOT MARKED AS BEING AVAILABLE FOR RE-USE. The VACUUM command must be run to make space available for re-use or to remove records from db. The command

VACUUM table-name

marks the deleted or updated records as being available for re-use within the table. It does not lock the table. The command

VACUUM FULL table-name

attempts to remove deleted or updated records from the table and make the space reusable by other tables.

(10) In Informix, all UNIX users are known to the server. In PostgreSQL, all users must be created using the CREATEUSER command.

(11) Database level privileges differ between Informix and PostgreSQL.

- in psql, if user A creates a db, then user B automatically has access to it
  (note that users A and B must be known to psql through the CREATEUSER command)

- in Infx: if user A creates a db, then user B does NOT have access to it unless granted CONNECT, RESOURCE or DBA privilege by user A

Table level privileges also differ

- in psql, if user A creates a table, users have NO privileges on the table (unless they are granted)

- in Infx, if user A creates a table, all users have all privileges on the table (unless they are revoked)

(12) PostgreSQL does not implement "lock mode" or "extent" attributes for tables. They must be removed from PostgreSQL DDL statements.

(13) Load file formats differ slightly bet Informix and PostgreSQL.

if “|” is the delimiter:

Infx: field1| field2 | field3 |

psql: field1| field2 | field3 (no delimiter on end of line)

(14) The Informix data type "datetime …." is replaced by the data type "timestamp" in PostgreSQL.

(15) In Informix, a datetime value of "08-01" defined as "month to day" is valid. It is NOT a valid timestamp value in PostgreSQL.

(16) In PostgreSQL, if the "-t" option is not specified for ecpg and a EXEC SELECT statement fails, then all subsequent EXEC SELECT statements will fail. See the man page for ecpg for details.

(17) In Informix, the keyword "CURRENT" translated into the current date and time. In PostgreSQL, use the keyword "CURRENT_TIMESTAMP" for this purpose.